Low-cost SCADA/HMI with Tiny Machine Learning for Monitoring Indoor CO_2 Concentration

I Nyoman Kusuma Wardana*‡, Suhaib A. Fahmy†*, Julian W. Gardner*
*School of Engineering
University of Warwick
Coventry CV4 7AL, United Kingdom
kusuma.wardana@warwick.ac.uk, j.w.gardner@warwick.ac.uk
† King Abdullah University of Science and Technology
Thuwal 23955, Saudi Arabia
suhaib.fahmy@kaust.edu.sa
‡ Politeknik Negeri Bali
Bali 80361, Indonesia

***Abstract*—Concerns about indoor air pollution are increasing as individuals spend much of their time indoors, with carbon dioxide being a notable concern. The increasing interest in developing low-cost gas sensors for indoor air quality monitoring has led to a surge in air quality data generated by these sensing devices. This growing volume of data creates opportunities for implementing machine learning methods in air quality research. However, a challenge of these indoor sensing devices is their resource-constrained memory and computing capabilities, making deploying machine learning algorithms challenging. This paper explores integration of low-cost Supervisory Control and Data Acquisition (SCADA) with tiny machine learning (TinyML) for effective monitoring of current and prediction of future CO_2 concentrations. The trained predictors produce RMSE ranging from approximately 0.5–7 ppm when predicting future CO_2 concentrations 1, 15, and 30 minutes ahead. Moreover, the models consistently yield confident R^2 scores, ranging from approximately 0.8 to 0.99.**

***Index Terms*—SCADA/HMI, tinyML, machine learning, microcontroller, air pollution**

I. Introduction

The global significance of air pollution has escalated due to its adverse effects on human health and its impact on socio-economic activities [1]. Factors such as population growth, industrial expansion, and economic development contribute to air pollution [2]. The effects of air pollution on the human body can vary, influenced by several factors, including the type of pollutant, duration of exposure, and quantity of exposure.

As individuals spend most of their time indoors [3]–[5], there is a rising concern about indoor air pollution and its potential health effects, especially for those working in such spaces [6]. Approximately 80–90% of individuals allocate their time indoors, such as in homes, schools, and offices [5]. Among the various indoor air contaminants reported, carbon dioxide (CO_2) contributes to this issue [7], and experts recommend paying attention to air pollutants to ensure the comfort and health of occupants in those spaces [8].

There is growing interest in developing low-cost gas sensors for various applications, including indoor air quality monitoring [9]. With the increasing volume of air quality data generated by these sensing devices, implementing machine learning methods holds promising potential for performing tasks such as classification, anomaly detection, and prediction of indoor air contaminants. However, since most indoor sensing devices have resource-constrained memory and computing capabilities, deploying machine learning algorithms becomes challenging for these particular devices. Hence, a novel paradigm called Tiny Machine Learning (TinyML) is introduced as a viable option for running machine learning algorithms on endpoint devices. Machine learning developments are moving towards edge computing [10], and a recent survey provides insight into how developers drive innovation in TinyML [11].

Indoor air quality parameters can be monitored using various technologies, including implementing an industrial standard system called Supervisory Control and Data Acquisition (SCADA). SCADA systems are commonly used to monitor and control critical infrastructure components in industrial sectors [12], and typically include an interface that facilitates interaction with human operators, referred to as the human-machine interface (HMI) [13]. For small-scale applications, opting for a low-cost SCADA system can be advantageous in terms of cost efficiency. For example, Aghenta and Iqbal (2019) employed a cost-effective SCADA system, utilizing an ESP32 Thing microcontroller board as a remote terminal unit (RTU) and a Thinger.IO local server IoT platform as a master terminal unit (MTU). This setup allowed for remote monitoring of photovoltaic voltage, current, power, and storage battery voltage [14]. Osman *et al*. introduced a Remote SCADA System (RSS) designed to control high-power machines and gather data from installed sensors [15]. The RSS boasts improved system latency and security and is suitable for various application platforms. In their application scenario, ESP8266 microcontroller boards with WiFi communication were employed to read and write signals to the GPIOs of the boards.

In this study, we enhance low-cost SCADA functionality with TinyML, extending beyond monitoring to the prediction of indoor air quality. In the experimental setup, we utilize low-

TABLE I
DESCRIPTIVE STATISTICS OF THE DATASET FEATURES GATHERED FROM ROOM05

	T (°C)	H (%)	P (hPa)	CO_2 (ppm)
count	16731	16731	16731	16731
mean	20.94	20.74	1004.63	413.43
std	0.33	4.95	15.01	16.65
min	20.47	8.44	977.40	391.00
25%	20.71	17.56	993.55	405.00
50%	20.84	23.06	1001.55	410.00
75%	21.19	24.09	1013.10	417.00
max	22.62	26.00	1036.65	556.00

cost microcontrollers as remote terminal units and a laptop as a master terminal unit. The connection between the RTUs and the MTU is established through the OPC protocol using the Ethernet. Given the resource constraints of the microcontrollers, we opt for linear regression as the machine learning algorithm to forecast future CO_2 concentration.

II. PROPOSED APPROACH

A. Data Source

For our study, we utilize the publicly available dataset provided by Räsänen *et al*. [16], which serves as supplementary material for the work published by Kallio *et al*. [17]. We adhere to the preprocessing procedure outlined by Kallio *et al*., yielding a refined dataset with measurements taken at 1-minute intervals. The features recorded during these measurements include CO_2 concentration, temperature (T), humidity (H), air pressure (P), and motion detection (PIR), all obtained from 13 different rooms.

The rooms consist of 11 office cubicles and two meeting rooms. For the scope of this study, three specific rooms (labelled as room05, room09, and room10) were chosen from the original dataset, as they exhibited the highest volume of data collection. Subsequently, all device timestamps were standardized from 2019-12-20 09:09:00 to 2019-12-31 23:59:00, resulting in 16,731 data points for each selected room. Moreover, this work excludes the motion sensor measurement, resulting in CO_2, temperature, humidity, and air pressure as the input features for the tinyML models.

Table. I presents the descriptive statistics of the dataset features obtained from room05, excluding the PIR feature. The mean CO_2 concentration measured during the specified period is approximately 413 ppm, with recorded minimum and maximum concentrations of 391 ppm and 556 ppm, respectively. Relatively minor fluctuations are observed in temperature, ranging from 20.47°C to 22.62°C. The computed standard deviation for humidity is 4.95%, indicating the extent to which the measured values deviate from the mean values. The CO2 concentrations in other rooms exhibit a notable similarity, with average values of approximately 417 ppm and 413 ppm for room09 and room10, respectively. Temperature and humidity in these rooms demonstrate relatively consistent means compared to those in room05.

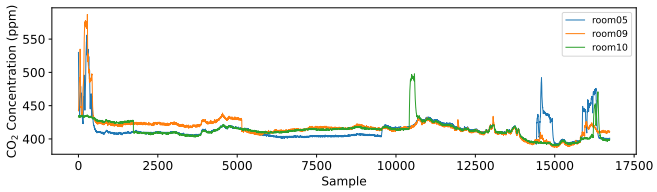

Fig. 1. Concentration of CO_2 in three different rooms.

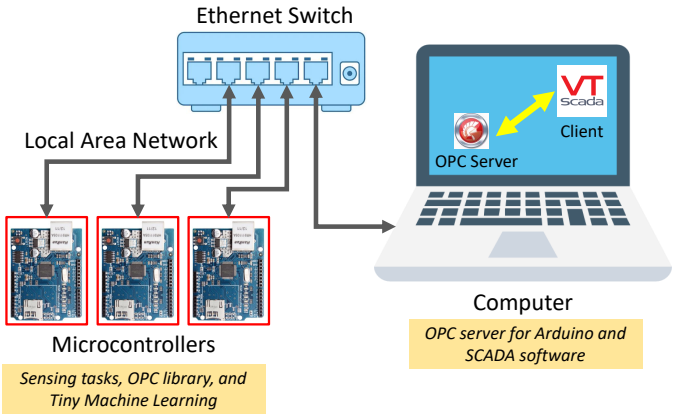

Fig. 2. Scenario of the SCADA/HMI system.

Fig. 1 illustrates the CO_2 concentrations measured in the three rooms. This study employs the proposed machine learning models to predict the last 20% of the CO_2 values. The training data constitutes the initial 80% of CO_2 concentration, along with air temperature, humidity, and air pressure.

B. Sensing Devices and SCADA System

This study employs three Arduino Uno microcontroller boards as RTUs, representing a distinct sensing device in each room. The Arduino Uno is an entry-level 8-bit microcontroller with a modest 2 kB of SRAM and 32 kB of flash memory. We utilized the Arduino OPC Server library provided by Martinez [18] to implement the SCADA communication protocol between the hardware and software sections. The microcontrollers established communication with the SCADA software through the OPC protocol, recognized as the interoperability standard for secure and reliable data exchange in industrial automation. The SCADA software chosen for this research is VTScadaLIGHT, developed by Trihedral [19].

The experimental setup for this study is illustrated in Fig. 2. As depicted, the key components consist of a computer, an Ethernet switch/hub, and microcontrollers with Ethernet shields. The computer and microcontrollers are linked to a Local Area Network via an Ethernet switch. The microcontrollers execute the sensing tasks, the OPC library, and the tiny machine learning algorithm, while the computer executes the SCADA software and a lightweight OPC server for Arduino. In this study, the laptop computer is equipped with an Intel® Core™ i7-8565U Processor and has 16 GB of RAM. The

operating system employed is Microsoft Windows 10 Home Edition.

C. Proposed TinyML Model

Various machine learning methods are applicable to regression problems, including linear regression, decision trees, random forests, support vector regression, neural networks, and more. The neural network is a widely employed method in TinyML. Nonetheless, there are alternative machine learning techniques viable for implementation in microcontrollers. Several of these techniques are more compact and faster to execute than neural networks while maintaining high accuracy across various tasks. Nevertheless, due to the constraints of the Arduino Uno (with only 2 kB of SRAM and 32 kB of flash memory), we consider linear regression to be the most efficient method for implementing tiny machine learning on resource-constrained microcontrollers.

The Ordinary Least Squares (OLS) Linear Regression (LR) implemented in this study adheres to the following general form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n \quad (1)$$

where y is the prediction output, β_0 is intercept, $\beta_1 \ldots \beta_n$ are the linear regression coefficients, and $x_1 \ldots x_n$ is the model inputs.

In this study, we deployed multiple predictors integrated into a single microcontroller. The original data comprises measurements at 1-minute intervals, and we provided predictions for three periods: 1-minute, 15-minute, and 30-minute ahead. The current CO_2 and prediction concentrations are displayed on the SCADA interface.

III. Results and Discussion

A. Experimental Setup

The experimental arrangement shown in Fig. 3 includes a laptop computer, microcontrollers equipped with ethernet shields, an ethernet switch/hub, and a power source. The laptop (MTU) serves as the host for the SCDA/HMI software and OPC server for Arduino. In contrast, the microcontrollers (RTUs) function as hosts for reading of sensor data, execution of the linear regression algorithm, and facilitation of OPC communication between the MTU and RTUs.

In this experiment, we mimick real sensor readings using the aforementioned sensor dataset on each microcontroller, stored in an external SD memory card. Throughout the experiments, data from each microcontroller are read every minute. Subsequently, once the input data are retrieved from the SD card, they are forwarded to three distinct linear regressors. The current values of CO_2, temperature, humidity, and predicted CO_2 are then transmitted to the SCADA/HMI. The specific identity of each value intended for transmission to the SCADA software is represented by tags.

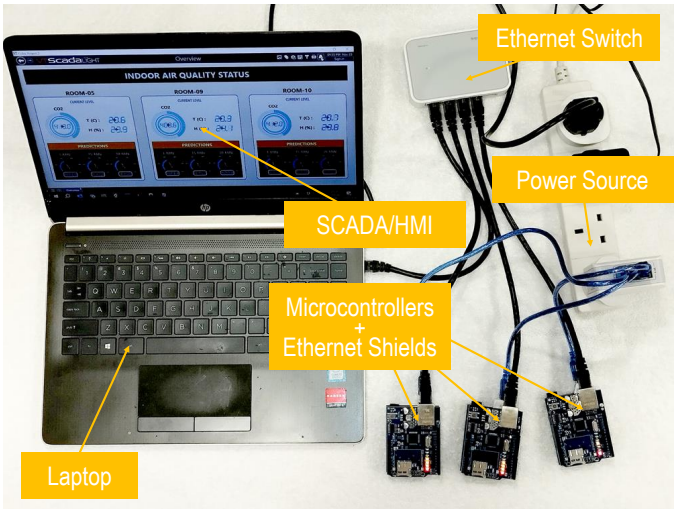

Fig. 3. Experimental setup of MTU and RTUs

B. SCADA/HMI Realization

Fig. 4 presents the design of the SCADA interface employed in this study. The interface presents the measurement of current and predicted data for the three rooms. In the upper section of each room, the interface displays the current CO_2 concentration, temperature, and humidity values. The air pressure value is omitted from the interface display to minimize the utilization of tags. The lower section of each room displays the forecasted CO_2 concentrations for various time intervals: 1 minute ahead, 15 minutes ahead, and 30 minutes ahead.

C. TinyML Evaluation

Three predictors are integrated into each microcontroller. The linear model intercept and coefficients are detailed in Table II, and these values adhere to the equation presented in Equation 1. Given the presence of four input features, there are four linear model coefficients and one intercept. To illustrate, in room05, when predicting CO_2 one minute ahead, the following equation is applied:

$$y = 2.4274 - 0.0058x_1 - 0.0052x_2 - 0.0006x_3 + 0.9961x_4 \quad (2)$$

Another example, when predicting CO_2 30 minutes ahead in room10, we use the following equation:

$$y = -76.3592 - 6.1471x_1 + 0.3897x_2 + 0.2189x_3 + 0.9431x_4 \quad (3)$$

Where y is the predictor output, x_1 is the current temperature value, x_2 is the current humidity value, x_3 is the current air pressure value, and x_4 is the current CO_2 concentration.

The performance of each predictor is assessed using root mean squared error (RMSE). The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \quad (4)$$

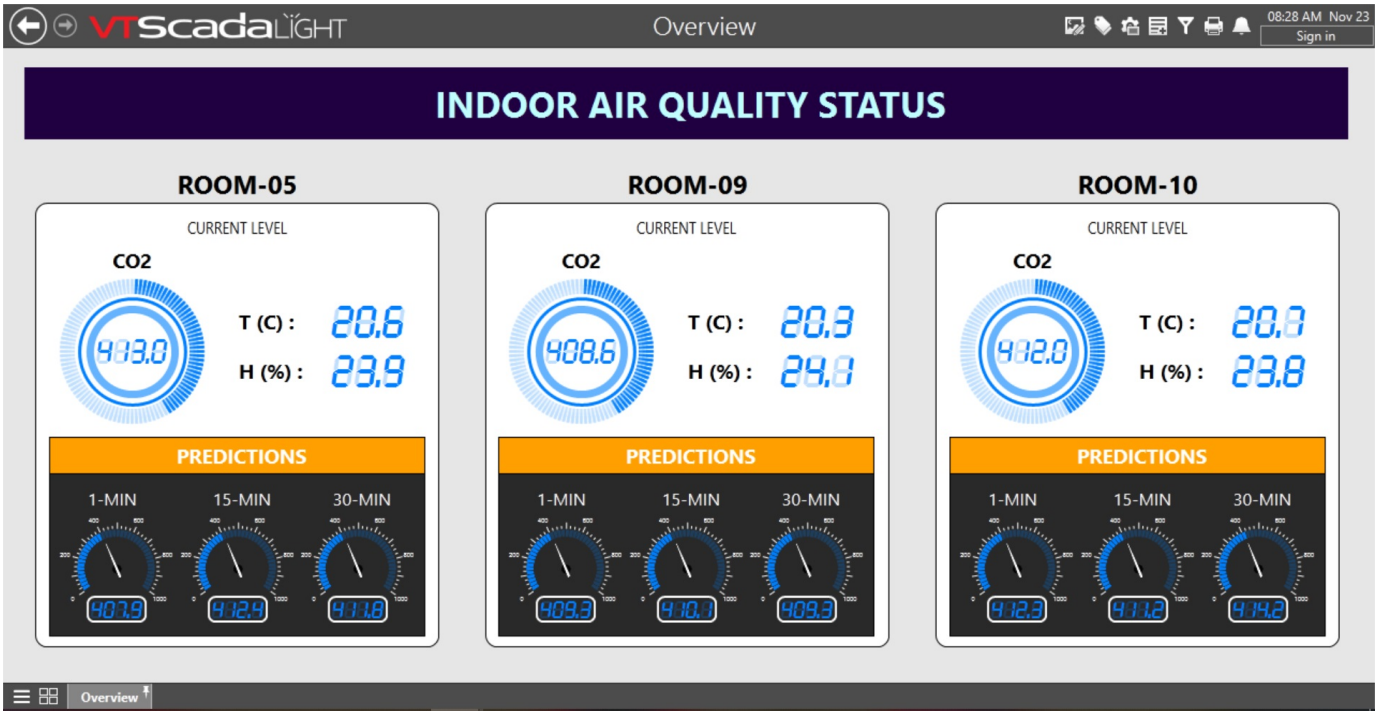

Fig. 4. SCADA interface for indoor air quality monitoring

TABLE II
SELECTED LINEAR REGRESSION COEFFICIENTS OF THE TINYML MODELS

	β_0	β_1	β_2	β_3	β_4
room05					
1-min	2.4274	−0.0058	−0.0052	−0.0006	0.9961
15-min	82.0247	1.2690	−0.2529	−0.0678	0.9150
30-min	160.4359	2.7769	−0.5089	−0.1398	0.8372
room09					
1-min	1.1177	0.0317	−0.0034	−0.0009	0.9981
15-min	30.0546	0.7259	−0.0933	−0.0264	0.9608
30-min	87.6559	1.9713	−0.2693	−0.0783	0.8957
room10					
1-min	−1.9831	−0.1457	0.0098	0.0055	0.9983
15-min	−38.0419	−2.7976	0.1847	0.1002	0.9801
30-min	−76.3592	−6.1471	0.3897	0.2189	0.9431

where n is the total number of data samples, y_i is the actual CO_2, and $\hat{y}_i$ is the predicted CO_2.

Using a laptop computer, we performed repeated 5-fold cross-validation to train and test the linear regressors to obtain less biased results. Each fold was reiterated 10 times with varying random seeds, and subsequently, the boxplot of the root mean squared error (RMSE) was generated, as shown in Fig. 5.

As depicted in Fig. 5, an extended prediction for CO_2 concentration leads to increased prediction errors. For instance, in room05, the average RMSE values are 0.82 ppm, 5.12 ppm, and 7.25 ppm for predicting 1 minute, 15 minutes, and 30 minutes, respectively.

The average R^2 value for each cross-validation iteration can

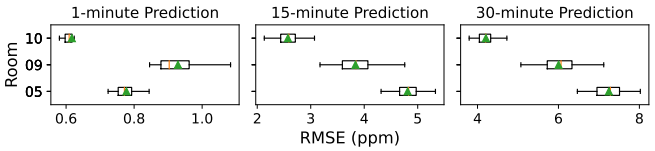

Fig. 5. Model performance evaluated using RMSE

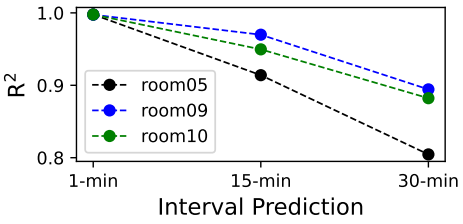

Fig. 6. Model performance evaluated using average R^2

be also determined by applying the following equation:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{5}$$

where n is the total number of data samples, y_i is the actual CO_2, $\hat{y}_i$ is the predicted CO_2 concentration, and $\bar{y}$ is the overall mean of the actual CO_2 concentration.

Illustrated in Fig. 6, longer-term predictions yield decreased R^2 values. Nevertheless, the models can predict future CO_2

concentrations with robust R^2 scores, ranging from approximately 0.8 to 0.99.

IV. Conclusion

This paper demonstrated integration of tiny machine learning for microcontrollers with low-cost SCADA/HMI to monitor and predict indoor CO_2 concentrations. Due to resource constraints in the microcontrollers used, a simple linear regression model was chosen as the machine learning algorithm. The results indicate that the SCADA/HMI software effectively displays both current and future CO_2 concentrations. Evaluated using the RMSE metric, the linear regression models produce error values ranging from approximately 0.5 ppm to 7 ppm when predicting future CO_2 concentrations (1 minute, 15 minutes, and 30 minutes ahead). The models consistently yield confident R^2 scores, ranging from approximately 0.8 to 0.99.

Acknowledgment

This work was partly supported by the Indonesia Endowment Fund for Education (LPDP), Ministry of Finance, Republic of Indonesia, under grant number Ref: S-1027/LPDP.4/2019.

References

[1] S. Dhingra, R. B. Madda, A. H. Gandomi, R. Patan, and M. Daneshmand, "Internet of things mobile–air pollution monitoring system (IoT-mobair)", IEEE Internet Things J., vol. 6, no. 3, pp. 5577–5584, Jun. 2019.

[2] L. Prieto-Parra, K. Yohannessen, C. Brea, D. Vidal, C. A. Ubilla, and P. Ruiz-Rudolph, "Air pollution, PM 2.5 composition, source factors, and respiratory symptoms in asthmatic and nonasthmatic children in Santiago, Chile", Environ. Int., vol. 101, pp. 190–200, Apr. 2017.

[3] Y. Sun, J. Hou, R. Cheng, Y. Sheng, X. Zhang, and J. Sundell, "Indoor air quality, ventilation and their associations with sick building syndrome in Chinese homes", Energy Build., vol. 197, pp. 112–119, Aug. 2019.

[4] C. Schweizer *et al.*, "Indoor time-microenvironment-activity patterns in seven regions of Europe", J. Expo. Sci. Environ. Epidemiol., vol. 17, no. 2, pp. 170–181, Mar. 2007.

[5] O. A. Postolache, J. M. D. Pereira, and P. M. B. S. Girao, "Smart sensors network for air quality monitoring applications", IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 3253–3262, Sep. 2009.

[6] S. Taheri and A. Razban, "Learning-based CO2 concentration prediction: Application to indoor air quality control using demand-controlled ventilation", Build. Environ., vol. 205, no. 108164, p. 108164, Nov. 2021.

[7] L. Zhang and F. Tian, "Performance study of multilayer perceptrons in a low-cost electronic nose", IEEE Trans. Instrum. Meas., vol. 63, no. 7, pp. 1670–1679, Jul. 2014.

[8] J. Saini, M. Dutta, and G. Marques, 'Indoor air quality monitoring with IoT: Predicting PM10 for enhanced decision support', in 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 2020.

[9] R. Ghosh, J. W. Gardner, and P. K. Guha, "Air pollution monitoring using near room temperature resistive gas sensors: A review", IEEE Trans. Electron Devices, vol. 66, no. 8, pp. 3254–3264, Aug. 2019.

[10] I. N. K. Wardana, S. A. Fahmy, and J. W. Gardner, "TinyML models for a low-cost air quality monitoring device", IEEE Sens. Lett., vol. 7, no. 11, pp. 1–4, Nov. 2023.

[11] Arm, "The Future of ML Shifts to the Edge", 2023. [Online]. Available: https://www.arm.com/markets/artificial-intelligence. [Accessed: 21-Nov-2023].

[12] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, 'A survey on SCADA systems: Secure protocols, incidents, threats and tactics', IEEE Commun. Surv. Tutor., vol. 22, no. 3, pp. 1942–1976, 2020.

[13] I. Qasim, M. W. Anwar, F. Azam, H. Tufail, W. H. Butt, and M. N. Zafar, "A model-driven mobile HMI framework (MMHF) for industrial control systems", IEEE Access, vol. 8, pp. 10827–10846, 2020.

[14] L. O. Aghenta and M. T. Iqbal, 'Low-cost, Open Source IoT-based SCADA system design using Thinger.IO and ESP32 Thing', Electronics (Basel), vol. 8, no. 8, p. 822, Jul. 2019.

[15] F. A. Osman, M. Y. M. Hashem, and M. A. R. Eltokhy, 'Secured cloud SCADA system implementation for industrial applications', Multimed. Tools Appl., vol. 81, no. 7, pp. 9989–10005, Mar. 2022.

[16] P. Räsänen *et al*., "VTT SCOTT IAQ Dataset". Zenodo, 2020, https://zenodo.org/records/4311286.

[17] J. Kallio, J. Tervonen, P. Räsänen, R. Mäkynen, J. Koivusaari, and J. Peltola, "Forecasting office indoor CO2 concentration using machine learning with a one-year dataset," Build. Environ., vol. 187, no. 107409, p. 107409, Jan. 2021.

[18] I. Martinez, "Software tools for makers," Software Tools for Makers, https://www.st4makers.com/ (accessed Nov. 13, 2023).

[19] Trihedral, VTScada by Trihedral, https://www.vtscada.com/ (accessed Nov. 13, 2023).